

AMENDMENTS

In the Claims

1. (Currently Amended) A method of determining an execution order for machine instructions to reduce spill code, said method comprising the step of:
from machine instructions that are ready for scheduling, scheduling the machine instruction for which an amount by which a size of a committed set of machine instructions would increase upon the scheduling of said machine instruction is smallest; and,
determining an execution order for the machine instructions to reduce spill code.
2. (Original) The method of claim 1, wherein said committed set of machine instructions includes any machine instruction that is already scheduled and any machine instruction that is descendent from an already scheduled machine instruction.
3. (Original) The method of claim 2, wherein, for each of said machine instructions ready for scheduling, said amount is determined by: identifying descendent machine instructions of each of said machine instructions; and determining which of said descendent machine instructions and said machine instructions is not in said committed set of machine instructions.
4. (Original) The method of claim 1, wherein said committed set of machine instructions includes any machine instruction that is descendent from an already scheduled machine instruction.
5. (Original) The method of claim 4, wherein, for said each machine instruction ready for scheduling, said amount is determined by:
identifying descendent machine instructions of each of said machine instructions; and
determining which of said descendent machine instructions is not in said committed set of machine instructions.

6. (Original) The method of claim 1, wherein a given machine instruction is considered ready for scheduling when scheduling of said given machine instruction as a next machine instruction would not cause an erroneous programmatic result.

7. (Original) The method of claim 1, wherein said method is undertaken when a risk of register overcommittedness exceeds a certain threshold.

8. (Original) The method of claim 7, wherein said threshold is exceeded when processor register availability drops below a particular threshold.

9. (Original) A method of determining an execution order for machine instructions to reduce spill code, said method comprising the steps of:

in a first bit vector containing one bit to represent each machine instruction to be scheduled, setting those bits for which the represented machine instruction is not committed, and resetting the remaining bits;

for said each machine instruction to be scheduled that is ready for scheduling:

in a second bit vector also having one bit to represent each machine instruction to be ordered in the same sequence as in said first bit vector, setting those bits for which the represented machine instruction is a descendant of said each machine instruction that is ready for scheduling, and resetting the remaining bits;

performing a bitwise AND operation of said first bit vector and said second bit vector to create a third bit vector; and

determining the number of set bits in said third bit vector; and

selecting for execution the machine instruction for which said third bit vector contains a minimum number of set bits.

10. (Original) The method of claim 9, said method further comprising the step of, prior to performing said bitwise AND operation, setting in said second bit vector the bit for which the represented machine instruction is said each machine instruction that is ready for scheduling.

11. (Currently Amended) A computer program product having media including computer programmed instructions for determining an execution order for machine instructions to reduce spill code, said computer program product comprising:

first subprocesses for, from machine instructions that are ready for scheduling,
scheduling the machine instruction for which an amount by which a size of a committed set of machine instructions would increase upon the scheduling of said machine instruction is smallest and determining an execution order for the machine instructions to reduce spill code.

12. (Original) The computer program product of claim 11, wherein said committed set of machine instructions includes any machine instruction that is already scheduled and any machine instruction that is descendent from an already scheduled machine instruction.

13. (Original) The computer program product of claim 12, said computer program product further comprising:

second subprocesses, wherein, for each of said machine instructions ready for scheduling, said amount is determined by:
identifying descendent machine instructions of each of said machine instructions;
and
determining which of said descendent machine instructions and said machine instructions is not in said committed set of machine instructions.

14. (Original) The computer program product of claim 11, wherein said committed set of machine instructions includes any machine instruction that is descendent from an already scheduled machine instruction.

15. (Original) The computer program product of claim 14, said computer program product further comprising:

second subprocesses, wherein, for said each machine instruction ready for scheduling, said amount is determined by:
identifying descendent machine instructions of each of said machine instructions;
and

determining which of said descendent machine instructions is not in said committed set of machine instructions.

16. (Original) A system in a computing environment for determining an execution order for machine instructions to reduce spill code, said system comprising:

first means for, in a first bit vector containing one bit to represent each machine instruction to be scheduled, setting those bits for which the represented machine instruction is not committed, and resetting the remaining bits;

second means for, for said each machine instruction to be scheduled that is ready for scheduling:

in a second bit vector also having one bit to represent each machine instruction to be ordered in the same sequence as in said first bit vector, setting those bits for which the represented machine instruction is a descendant of said each machine instruction that is ready for scheduling, and resetting the remaining bits;

performing a bitwise AND operation of said first bit vector and said second bit vector to create a third bit vector; and

determining the number of set bits in said third bit vector; and

third means for selecting for execution the machine instruction for which said third bit vector contains a minimum number of set bits.

17. (Original) The system of claim 16, said system further comprising:

fourth means for, prior to performing said bitwise AND operation, setting in said second bit vector the bit for which the represented machine instruction is said each machine instruction that is ready for scheduling.